

Testovi prihvatljivosti i razboritosti

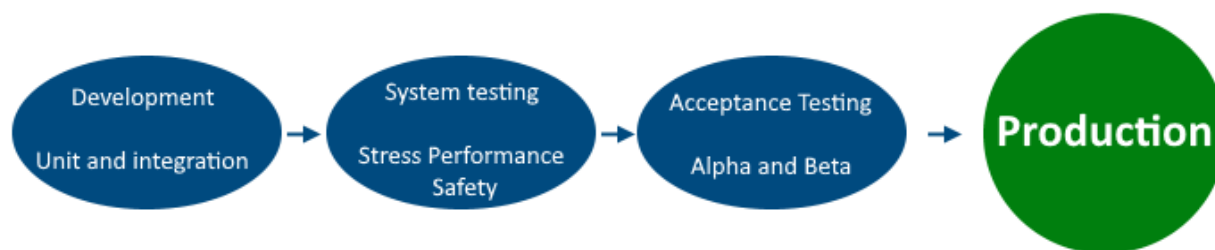
U okviru ove lekcije govorit ćemo o posebnom vidu nefunkcionalnih testova – testovima koje u velikoj mjeri sprovode korisnici. Testovi ovoga tipa su testovi prihvatljivosti, a dva najznačajnija vida testiranja u okviru testiranja prihvatljivosti su alpha i beta testovi.

Kao što smo naveli, kada se završi funkcionalno testiranje i testiranje performansi, naručilac preuzima ključnu ulogu, vodi testiranje i definiše slučajeve koji će se testirati. Svrha testa prihvatanja jeste da se kupcima i korisnicima omogući da utvrde da li sistem zaista zadovoljava njihove potrebe i očekivanja.

Pre nego što nastavimo dalje, važno je napomenuti da pored alpha i beta testiranja, koja predstavljaju primarni pristup testiranju u ovoj fazi, postoje i neki testovi koji se ne tiču direktno razvoja softvera i samog koda, ali i dalje pripadaju ovoj fazi, pa ćemo i njih ukratko spomenuti:

- Business acceptance testing (BAT) – U okviru ovog testiranja vrši se provera da li program odgovara samim ciljevima sa biznis aspekta – koliko će uticati na profit; da li prati promene tržišta i napredovanje tehnologije. Proizvod koji potpuno ispunjava sve tehničke zahteve može pasti na ovome testu upravo ako ne ispuni neki od navedenih kriterijuma.
- Contract acceptance testing (CAT) – Ovaj test predstavlja vid ugovora u kojem se jasno navodi koje stavke inicijalnog dogovora dolaze do krajnjeg korisnika. Takođe, koje je to testove prihvatljivosti kompanija koja razvija softver dužna da sprovede pre dostavljanja softvera i naplate rada. U ugovoru se navode tačni periodi testiranja, oblasti testiranja i uslovi. Ovo se ujedno naziva testom, jer se ovaj inicijalni dokument koristi kao test ispunjenja uslova i dogovora u kasnijim fazama testiranja.
- Regulations/compliance acceptance testing (RAT) – Ovim testom se utvrđuje da li softver krši pravila vlade u državi gde će se softver primenjivati. RAT test se često izvršava kada organizacija sarađuje sa stranim partnerom. Kao i kod BAT testa, iako je program tehnički ispravan i ispunjava sve zahteve ugovora, ukoliko ne radi prema pravilima države u kojoj će se primenjivati, on se ne može koristiti. U poslednjim godinama odličan primer upravo potrebe za nekim vidom RAT testiranja je uvođenje pravilnika GDPR o zaštiti privatnosti i podataka korisnika na teritoriji Evropske unije. Uvođenje ove legislative je primoralo sve kompanije koje posluju na teritoriji EU da usklade svoje web aplikacije tako da poštuju ovu novu regulativu. Upravo RAT testiranjem utvrđujemo koliko poštujemo neki zakon ili odstupamo od njega.

Upravo smo iz ovih razloga u prethodnoj lekciji napomenuli da se testiranje ne sprovodi po striktnoj proceduri i testovima unutar koda; samo na osnovu prethodno navedenih testova, vidite da tu nisu zastupljeni klasični pristupi provere koda ili rada softvera, već finansija, pravila, poštovanja ugovora i slično. Sada ćemo se upoznati sa alpha i beta testiranjem – metodama testiranja programa koje se sprovode kao jedan od poslednjih nivoa testiranja.



Slika 11.1. Faze testiranja softvera

Alpha testiranje

Alpha testiranje je tip softverskog testiranja koje se izvršava u cilju pronalaženja grešaka pre objavljivanja softvera realnim korisnicima, tj. javnosti. Kao što smo naveli, po tipu je vid testiranja prihvatljivosti softvera. Testiranje nosi naziv *alpha testiranje* jer se obavlja pred kraj ciklusa razvoja softvera.

Alpha testiranje se, u zavisnosti od toga ko je vlasnik i naručilac softvera, može izvršavati na dva načina:

1. Ukoliko je kompanija u kojoj razvojni tim radi ujedno i vlasnik samog softvera, tada alpha testiranje sprovedu članovi razvojnog tima, osposobljeni za obezbeđenje kvaliteta softvera.
2. Ukoliko se softver radi za drugu kompaniju, alpha testiranje organizuje naručilac, čiji zaposleni testiraju softver i uveravaju se da sve radi kao što je dogovoreno.

Sve verzije softvera kreirane u ovoj fazi testiranja deklariraju se kao alpha verzije softvera. Ove verzije, naravno, podrazumevaju završene celine programa, koje se sada ažuriraju sa ispravkama napravljenim u ovoj fazi.

Kako izgleda proces alpha testiranja

Proces alpha testiranja možemo sagledati kao proces koji se sastoji iz dve faze. U prvoj fazi in-house razvojni tim koristi interno razvijene alate za simulaciju radnog okruženja programa; koriste se razne metode da bi se detaljno ispitao rad programa. U prvoj fazi se otkrije najveći broj grešaka i funkcionalnosti koje nedostaju.

Drugu fazu alpha testiranja izvršava osoblje zaduženo za obezbeđenje kvaliteta. Dakle, unapređene alpha verzije programa sada dolaze do njih i vrši se dodatno testiranje programa. Ova faza takođe nema precizno određene testove; koriste se i black box i white box pristupi testiranju.

Drugu fazu alpha testiranja možemo zamisliti kao korišćenje programa koji nije spreman za objavljivanje, ali mu se otvara pristup da bi se dobile inicijalne povratne informacije od ljudi koji nisu učestvovali u njegovom razvoju.

Uobičajeno, prva faza alpha testiranja se obavlja na serverskom test okruženju. Dakle, uslovi za program su prethodno definisani, ne postoji opterećenje nad serverom po broju korisnika, ne postoje različita okruženja koja mogu uticati na rad programa i slično. Razvojni tim definiše ciljeve alpha testiranja i proverava ispunjenost tih ciljeva. U ovoj fazi razvojni tim unosi podatke u softver, prati njihovo kretanje, prati dobijene rezultate i generalno koristi softver.

U ovoj fazi se program ne testira na ostale faktore, jer je i dalje to prototip koji radi u test okruženju. Stoga se u ovoj fazi ne sprovodi testiranje pouzdanosti ni instalacija programa na druga okruženja, pa čak se ni preciznost dokumentacije programa ne proverava. Dakle, fokus je na pronalaženju grešaka u kodu koje su prošle integracione faze softvera.

Ono što je važno za organizaciju je definisanje test plana i test slučajeva, o kojima smo govorili na samom početku kursa. Dakle, precizno se definišu koraci koji su izvedeni u softveru, šta se dogodilo, a šta je trebalo da se dogodi, sve u svrhu evidencije grešaka. Ova faza se može ponoviti i nekoliko puta kroz više iteracija dok se ne dobije odgovarajući rezultat.

Druga faza, iako deluje slična prvoj, suštinski nije. U drugoj fazi testiranje više nije potpuno fokusirano na funkcionalno testiranje. Pored ponavljanja nekih funkcionalnih testova iz prve faze, QA tim kontroliše i druge faktore. Konkretno, koliko je lako koristiti softver, koliko je on vizuelno prijatan, koliko je lako kretati se kroz opcije, da li je brzina zadovoljavajuća i slično. Ovo se radi jer sledeći korak je upravo beta testiranje, koje se ne odigrava u test okruženjima, već program nakon uspešno završenog alpha testiranja kreće u primenu, gde prethodno spomenuti faktori postaju bitni.

Za ovu fazu testiranja programa postoje, naravno, uslovi koje je potrebno ispuniti pre početka, kao i uslovi koji moraju biti ispunjeni nakon završetka ove faze. Pa pogledajmo kako ti uslovi mogu izgledati.

Ulazni kriterijumi alpha testiranja:

- postoje dokumenta koja opisuju zahteve koje program mora da ispuni pre objavljivanja;
- kreirani su svi test slučajevi za prethodno definisane zahteve;
- tim koji testira alpha verzije programa upoznat je sa samim softverom;
- test okruženje obezbeđeno za ovu fazu (server, alati, drugi softver koji radi sa programom);
- postoji alat u kojem će se vršiti evidencija testiranja (najčešće druga web aplikacija koja služi za unos podataka).

Izlazni kriterijumi alpha testiranja:

- svi test slučajevi izvršeni i na kraju procesa označeni kao uspešni;
- svi problemi pronađeni u ovoj fazi ispravljeni;
- postignut dogovor da neće biti daljeg proširenja funkcionalnosti softvera;
- dostavljen izveštaj o alpha testiranju;
- zatvoreno alpha testiranje;
- priprema softvera za beta testiranje.

Ono što je još potrebno spomenuti kod alpha testiranja jesu neki pojmovi sa kojima su se najverovatnije mnogi od vas već susreli, a naročito oni koji prate industriju razvoja video-igara. U okviru ove industrije, a i mnogih drugih, često možete da čujete za termine *pre-alpha* i *early access alpha*. Spomenuto jesu alpha verzije programa, koje imaju jedan izuzetak: dostupne su korisnicima pre nego što zvanično uđu u alpha fazu testiranja.

Razloga za uvođenje ovih ranijih verzija programa ima više, od dobijanja dodatnog finansiranja projekta jer korisnici ranije plaćaju za softver, do ranijeg feedbacka o proizvodu. Čak i u slučaju open source programa, pre-alpha verzije programa pospešuju razvoj programa, jer veća zajednica programera može da doprinosi razvoju.

Beta testiranje

Beta testiranje je poslednja faza testiranja pre nego što program objavi i predstavi široj publici. Kao i kod alpha testiranja, cilj je otkrivanje što većeg broja grešaka i problema u korišćenju programa, dok još postoji u određenoj meri kontrolisano okruženje i broj korisnika.

U okviru ove faze, testiranje izvršavaju pravi korisnici u realnom okruženju, gde će koristiti svoje računare, mrežu itd. Beta testiranje predstavlja i fazu u kojoj se prvi put obavljaju testovi pouzdanosti, izdržljivosti i bezbednosti aplikacije, jer, kao što smo ranije naveli, ove testove je teško obaviti u test okruženju, sa malim brojem korisnika. Dakle, za razliku od alpha testiranja, beta testiranje je black box orijentisana faza u kojoj razvojni tim ne sprovodi direktno testiranje, već prati feedback korisnika i na osnovu njega pravi izmene i dopune programa.

Tipovi beta testiranja

Za razliku od alpha testiranja, u okviru beta testiranja postoji više različitih tipova:

- **Tradicionalno beta testiranje** predstavlja osnovni tip testiranja: program se primenjuje u realnim uslovima na ciljnoj grupi. Prikupljaju se podaci i na osnovu prikupljenih podataka se vrši unapređenje proizvoda.
- **Javno beta testiranje** – Program se objavljuje bez ciljne grupe i na taj način se dobija feedback o kvalitetu proizvoda. Jedan od najvećih programa testiran kroz public beta testiranje je bio Microsoftov operativni sistem Windows 8.
- **Marketing beta testiranje** – Cilj realizacije beta testiranja na ovaj način jeste privlačenje medija i budućih korisnika. Tokom ove faze pronalaze se najbolji kanali za promovisanje proizvoda, ali se ona ujedno koristi i za prikupljanje povratnih informacija o novoj funkcionalnosti ili proizvodu. Društvena mreža Instagram koristi marketing beta testiranje za promociju svojih novih opcija platforme, tako što angažuje najpopularnije članove u aktivnom promovisanju funkcije.
- **Tehničko beta testiranje** – Ovo je specifičan vid beta testiranja, gde se za ciljnu grupu biraju korisnici koji imaju iskustvo rada sa programima sličnog tipa. Na ovaj način se prilikom testiranja mogu otkriti neki dublji problemi programa ili nedostatak potrebne funkcionalnosti.
- **Fokusno beta testiranje** – Program se objavljuje, ali u cilju testiranja samo jedne od funkcionalnosti programa, najčešće glavne funkcionalnosti.
- **Post-release beta** – Verzija programa se objavljuje na tržište i podaci se prikupljaju samo u svrhe daljeg unapređenja i održavanja softvera, bez trenutnih izmena.

Kao što možete videti, beta testiranje se ne menja u ovim različitim tipovima – samo se menja cilj šta tačno želimo da postignemo beta testiranjem.

Pitanje

Testiranje kojim se utvrđuje da li softver krši pravila vlade u državi gde će se softver primenjivati nosi naziv:

- **RAT**
- CAT
- BAT

Objašnjenje:

Regulations/compliance acceptance testing (RAT) je vid testiranja kojim se utvrđuje da li softver krši pravila vlade u državi gde će se softver primenjivati. RAT test se često izvršava kada organizacija sarađuje sa stranim partnerom.

Closed i open beta testiranje

Koji god tip beta testiranja da izaberemo, postoje dva načina na koje ga možemo realizovati: otvorenim ili zatvorenim pristupom.

Closed beta testiranje predstavlja prikazivanje closed beta verzije programa. Razlog zašto se naziva closed jeste to što se prikazuje samo odabranim ljudima. Čest primer upotrebe ovog načina beta testiranja prati sledeći tok: kompanija kreira web sajt, često u vidu landing stranice za promovisanje proizvoda. Na sajtu ovog tipa postoji opcija da korisnici popune prijavu za novosti o razvoju softvera. Na ovaj način već postoji baza korisnika koji su kandidati za closed beta testiranje, jer su već zainteresovani za proizvod. Najbolji primer ove strategije je kompanija Apple, koja aktivno, kako za svoje uređaje tako i za softver, ima closed beta program: <https://beta.apple.com/sp/betaprogram/>.



Apple Beta Software Program

Help make the next releases of iOS, iPadOS, macOS, tvOS and watchOS our best yet. As a member of the Apple Beta Software Program, you can take part in shaping Apple software by test-driving pre-release versions and letting us know what you think.

Learn more about the next releases.

[iOS 14 >](#) [iPadOS 14 >](#) [macOS Big Sur >](#) [tvOS 14 >](#) [watchOS 7 >](#)

[Sign up](#)

Slika 11.2. Apple beta software program

Za razliku od closed beta testiranja, otvoreni tip, kako mu i samo ime sugeriše, ne ograničava pristup korisnicima. Open beta testiranje se najčešće obavlja nakon closed beta faze. Ovaj tip testiranja usled većeg broja korisnika omogućava prikupljanje veće količine kvantitativnih podataka, naročito po pitanju interakcije sa programom. Kompanija Google je tokom razvoja Gmail aplikacije provela pet godina (od 2004 do 2009) u beta fazi. Za to vreme konstantno je prikupljan feedback od korisnika i vreme je utrošeno da se isprave greške i da se aplikacija usavrši i spremi za objavljivanje. Tokom beta testiranja je zabeleženo preko 1,2 miliona aktivnih korisnika, ali čak ni ovo nije bio najveći beta test. Svoje mesto u istoriji najvećih beta testova imaju beta testiranje operativnog sistema Windows Vista i, u svetu video-igara, beta verzija popularne igre Minecraft.

Pogledajmo sada kako izgleda dijagram procesa jednog standardnog beta testa programa:



Slika 11.3. Dijagram procesa testiranja¹

Dakle, možete videti da postoji više faza. Broj faza, naravno, uvek zavisi od potreba projekta, ali suštinski, uvek krećemo sa time da jasno prepoznamo šta je potrebno kontrolisati beta testom i da razumemo koji su ciljevi koje treba ispuniti testiranjem. Nakon toga kreiramo dokumentaciju sa svim detaljima testova i organizujemo tim.

Sledeći važan korak je implementacija metodologije za praćenje problema, a nakon toga imamo više faza obrade dokumentacije i generisanja izveštaja beta testa. Kada govorimo o implementaciji metodologija za praćenje problema, nama važan aspekt su alati koji se koriste za beta testiranje.

¹ <https://www.indiamart.com/proddetail/beta-testing-6371620162.html>

Alati za beta testiranje

Dakle, kao što smo govorili do sada, proces beta testiranja se zasniva na povratnoj informaciji od korisnika. Stoga, potrebno je da imamo neki vid alata koji će nam služiti da beležimo koje su se greške javile i da generalno prikupljamo povratne informacije od korisnika tokom beta testiranja. Nama je važno da imamo alate ovoga tipa, jer oni olakšavaju proces beta testiranja.

Alati ovoga tipa su najčešće web aplikacije kojima korisnici mogu da pristupe i ostave svoje komentare na program. Na sledećim linkovima možete videti par primera kako je to realizovano u nekim kompanijama, kako iz gaming industrije tako i u oblasti poslovnog softvera:

- <https://rust.nolt.io/> – feedback beta faze video-igre Rust;
- <https://feedback.bistudio.com/tag/dayz/> – feedback beta faze video-igre DayZ;
- <https://nolt.io/> – web aplikacija Nolt, koja služi za organizaciju beta testa;
- <https://techcommunity.microsoft.com/t5/microsoft-edge-insider/ctp/MicrosoftEdgeInsider> – feedback beta faze za web pregledač Microsoft Edge.

Kao što možete videti, ovi primeri su vrlo slični u svojoj osnovi. U vidu su foruma, gde korisnici imaju opciju da glasaju za prijave grešaka ili sugestije za unapređenje. Dakle, najbitnija je funkcionalnost postavljanja komentara. Kompanije nekada izrađuju odvojene web sajtove za ovu namenu, a postoje i specijalizovani alati za slučaj da organizacija ne želi sama da razvija sama alat ovoga tipa:

- <https://buglife.com/>
- <https://instabug.com/>
- <https://www.bugsee.com/>

Navedeni alati, pored toga što omogućavaju korisnicima ostavljanje povratnih informacija, imaju mehanizme koji automatski generišu izveštaje o greškama. Takođe, imaju mogućnost praćenja statusa greške, pa čak i mogućnost obavljanja drugih testova.

Završni korak testa prihvatljivosti jeste obrada povratnih informacija dobijenih od korisnika. Svaki pojedinačni slučaj gde je neki korisnik ostavio komentar sada postaje test slučaj. Dakle, ukoliko se utvrdi da je to stvarno problem rada programa, prijavljena greška postaje test slučaj koji će biti integrisan u neki od testova nad kodom programa. Zadatak tima testera jeste da na osnovu povratne informacije korisnika zabeleže tačne korake korišćenja programa koji su izazvali grešku.

Često se generiše dokumentacija beta testa na osnovu podataka od korisnika; primer templejta ove dokumentacije možete videti na ovom [linku](#).

Za kraj ove lekcije, govorićemo o testiranju razboritosti, tj. jasnoće korišćenja i razumevanja rada programa. Predstavnik ove vrste testiranja je usability test.

Usability testiranje

Procesom usability testiranja se evaluira koliko je jednostavno za krajnje korisnike da savladaju i koriste softver, uključujući i korisničku dokumentaciju.

Ovaj test je zasnovan na realnim korisnicima i upotrebi programa. Poenta ovoga testa jeste da odgovori na pitanje: „Koliko su softverske funkcije efikasne u podršci korisničkim zadacima?”

Najčešće se odnosi na web sajtove i aplikacije, korisničke interfejsa i sl. Zbog fokusa na korišćenje programa, često se naziva i user experience (UX) test.

Ciljevi ovakvog testiranja su:

- efikasnost – koliko koraka je potrebno za kompletiranje zadatka, npr. naručivanje proizvoda iz online shopa;
- tačnost – koliko grešaka čine korisnici i koliko su one kritične;
- pamćenje – koliko korisnici pamte o samim koracima korišćenja nakon perioda nekorišćenja;
- emotivni utisak – kako se korisnik oseća nakon rada, da li je zadovoljan, pod stresom, da li bi preporučio softver nekome i sl.

Sa ovim ciljevima, često možemo i oformiti konkretna pitanja na koja je potrebno odgovoriti uspešnim testom, pa bi ta pitanja npr. mogla da glase:

Da li je sistem efikasan?

- Da li je sistem lak za savladavanje?
- Da li je sistem koristan i dodaje vrednost ciljnoj grupi?
- Da li je sadržaj poput boja, ikonica i slika vizuelno privlačan?

Da li je sistem efikasan za korišćenje?

- Nije potrebno mnogo navigacije da se dođe do željenih stranica i nema preterane potrebe za korišćenjem dodatne navigacije.
- Svi prikazi i stranice imaju uniforman izgled, što olakšava upoznavanje sa aplikacijom.
- Postoji opcija pretrage samog softvera ili web sajta.

Da li je program precizan i ažuran?

- Ne postoje zastareli ili netačni podaci na aplikaciji.
- Ne postoje linkovi koji ne rade.
- Kontrole su jasne i nema potrebe za dodatnom obukom za korišćenje programa.
- U okviru aplikacije postoji opcija pružanja pomoći u korišćenju.

Izvršavanjem procesa usability testa i dobijanjem povratnih informacija, omogućujemo unapređenje zadovoljstva krajnjih korisnika i činimo naš sistem efikasnijim prilikom korišćenja.

Rezime

- Pored alpha i beta testiranja, koja predstavljaju primarni pristup testiranju u ovoj fazi, postoje i neki testovi koji se ne tiču direktno razvoja softvera i samog koda, ali pripadaju ovog grupi. To su: BAT, CAT i RAT testovi.

- Alpha testiranje je tip softverskog testiranja koje se izvršava u cilju pronalaženja grešaka pre objavljivanja softvera realnim korisnicima, tj. javnosti. Po tipu je vid testiranja prihvatljivosti softvera. Testiranje nosi naziv alpha testiranje jer se obavlja pred kraj ciklusa razvoja softvera.
- Sve verzije softvera kreirane u alpha fazi testiranja deklariraju se kao alpha verzije softvera. Ove verzije, naravno, podrazumevaju završene celine programa, koje se sada osvežavaju sa ispravkama otkrivenim u ovoj fazi.
- Beta testiranje je poslednja faza testiranja pre nego što se program objavi i predstavi široj publici. Kao i kod alpha testiranja, cilj je otkrivanje što većeg broja grešaka i problema u korišćenju programa, dok još postoji u određenoj meri kontrolisano okruženje i broj korisnika.
- U okviru beta testa, testiranje izvršavaju korisnici u realnom okruženju, gde će koristiti svoje računare, mrežu itd. Beta testiranje predstavlja i fazu u kojoj se prvi put obavljaju testovi pouzdanosti, izdržljivosti i bezbednosti aplikacije.
- Procesom usability testiranja se radi procena koliko je jednostavno za krajnje korisnike da savladaju i koriste softver, uključujući i korisničku dokumentaciju. Poenta ovog testa jeste da odgovori na pitanje: „Koliko su softverske funkcije efikasne u podršci korisničkim zadacima?”
- Usability test se najčešće koristi za web sajtove i aplikacije, korisničke interfejsa i sl. Zbog fokusa na korišćenje programa, često se naziva i user experience (UX) test.

